

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií



**ústav nových technologií
a aplikované informatiky**

Bakalářská práce

Konfigurátor produktů jako komponenta CMS

Configuration tool for common CMS

Martin Hozák

Vedoucí bakalářské práce: Mgr. Jiří Vraný, Ph.D.

Konzultant Dis. Martin Hujsl

Studijní program: B2612 Elektrotechnika a informatika

Studijní obor: 1802R022 - Informatika a logistika

Květen 2011

Tady bude oficiální zadání.

Poděkování

Na tomto místě bych chtěl poděkovat Mgr. Jiřímu Vranému, Ph.D. za vedení bakalářské práce a za užitečné rady při psaní tohoto dokumentu. Mé poděkování patří také Dis. Martinu Hujslovi za rady a připomínky při tvorbě praktické části.

Největší poděkování však patří mým rodičům, za podporu během mého studia.

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

V Liberci dne

Podpis

ABSTRAKT

Tato práce se v první části zabývá možnostmi a způsoby komunikace prostřednictvím webApi rozhraní. Popisuje způsob komunikace prostřednictvím protokolu SOAP. Hlavní náplní je návrh modulu, který umožní prostřednictvím nabízeného API rozhraní vytvářet a spravovat dražby na portále Aukro.cz. Součástí je návrh jednoduchého systému, který prezentuje možnosti vytvořeného modulu. V závěru práce je popsán způsob implementace vytvořeného modulu do libovolného systému.

ABSTRACT

The first part of this presentation is mainly about potentials and methods communication through webApi interface. Describes communication through protocol SOAP. The project module is the main contents which enable create and manage auction on portal Aukro.cz through offered API interface. For demonstration of possibilities this module is designed a simple system. In the last step of this presentation is describes implementation module to the any system.

OBSAH

1. Úvod	11
2. Cíle bakalářské práce	12
3. Obecné principy	13
3.1. API rozhraní	13
3.2. Aukro API rozhraní	14
3.3. Tvorba komponent	14
4. Analýza, návrh řešení a výběr technologií	15
4.1. Současný stav	15
4.2. Návrh řešení	15
4.2.1. Návrh modulu	15
4.2.2. Návrh testovacího systému	16
4.3. Výběr technologií	16
4.3.1. XHTML	16
4.3.2. CSS	16
4.3.3. JavaScript	16
4.3.4. PHP	17
4.3.5. AJAX	17
4.3.6. MySQL	17
4.3.7. Použití open source	17
5. Realizace	19
5.1. Způsob realizace	19
5.2. Logika systému	19
5.2.1. Modul zboží	19
5.2.2. Modul objednávky	20
5.2.3. Modul úvod	20
5.3. Logika modulu Aukro	20

5.3.1.	Část 1 – nastavení modulu	21
5.3.2.	Část 2 – vystavení produktu.....	22
5.3.3.	Část 3 – seznam vystavených	23
5.4.	Adresářová struktura systému.....	25
5.5.	Adresářová struktura modulu aukro	26
5.6.	Souborová struktura systému	26
5.7.	Souborová struktura modulu.....	27
5.8.	Popis php skriptů.....	28
6.	Postup nasazení modulu	29
6.1.	Převzetí celé struktury	29
6.2.	Vkládání jednotlivých částí modulu.....	30
6.2.1.	Zprovoznění nastavení.....	30
6.2.2.	Zprovoznění vystavování aukcí	31
6.2.3.	Zprovoznění seznamu aukcí	31
6.3.	Oživení modulu.....	33
6.3.1.	Požadavky k systému Aukro.....	33
6.3.2.	Nastavení modulu.....	33
6.4.	Prezentace modulu	34
7.	Závěr.....	35
8.	Zdroje informací.....	36
	Příloha – A obsah přiloženého CD.....	37

Seznam zkratek

AJAX - Asynchronous JavaScript and XML

API - Application Programming Interface

CMS - Content Management System

CSS - Cascading Style Sheets

HTML – Hyper Text Markup Language

MySQL - My Structured Query Language

PHP - PHP: Hypertext Preprocessor

SQL - Structured Query Language

W3C - World Wide Web Consortium

DTD - Document Type Definition

BSD - Berkeley Software Distribution

GPL - General Public License

LGPL - Lesser General Public License

MPL - Mozilla Public License

WSDL - Web Service Definition Language

SOAP - Simple Object Access Protocol

XML - Extensible Markup Language

HTTP - Hypertext Transfer Protocol

Klíčová slova

modul, API, SOAP, Aukro, univerzální, aukce

Keywords

module, API, SOAP, Aukro, universal, auction

Seznam obrázků

Obrázek 1: schéma komunikace s rozhraním API	13
Obrázek 2: ukázka zprávy v XML jazyce	14
Obrázek 3: postup implementace FCKeditoru.....	18
Obrázek 4: postup implementace frameworku JAK	18
Obrázek 5: schéma propojení tříd modulu.....	21
Obrázek 6: struktura databázové tabulky	21
Obrázek 7: diagram vkládání skriptů v systému	28
Obrázek 8: ukázka komentáře.....	28
Obrázek 9: ukázka vložení nutných souborů do hlavičky.....	29
Obrázek 10: ukázka vytvoření stránky pro nastavení.....	30
Obrázek 11: ukázka změny adresy odkazu.....	30
Obrázek 12: ukázka implementace vystavování produktů	31
Obrázek 13: ukázka implementace seznamu vystavených produktů	31
Obrázek 14: ukázka skriptu vytvářející objednávky.....	32
Obrázek 15: ukázka nastavení přihlašovacích údajů.....	33
Obrázek 16: ukázka nastavení umístění obchodu	33
Obrázek 17: ukázka nastavení doprav	34

1. Úvod

V dnešní elektronické době je nakupování na internetu zcela běžné a pro někoho již jedinou alternativou. V elektronických obchodech lze zakoupit a nechat si dovést prakticky jakékoliv zboží až do domu. Seznam obchodů, které nabízejí své produkty se neustále rozrůstá a každý z nich chce pro sebe získat co největší kus trhu. Proto jsou kladeny velké nároky na technické řešení systémů, na kterých jsou internetové obchody provozovány. A společnosti zabývající se dodáváním těchto systémů musí svůj produkt neustále vylepšovat, aby si udržely své stávající zákazníky a nalákaly nové. Jedním z trendů pro nabízení svých produktů, je využívání aukčních serverů, jejichž obliba poslední dobou strmě stoupá. V České Republice působí velký aukční portál Aukro.cz, který nabízí své API rozhraní umožňující komunikaci s jinými systémy.

Nutnost neustále vylepšovat systém musí řešit i společnost CLWEB, která vyvíjí svůj vlastní CMS systém. Společnost působí převážně v severních Čechách a svým zákazníkům se snaží nabízet systém, který umožní dostat jejich obchod mezi špičku. Proto při stoupající oblibě aukčních systémů, chce zákazníkům umožnit prezentovat své produkty i touto cestou. A hlavně jim usnadnit práci.

Cílem práce je vyvinout modul, který půjde snadno zasadit do stávajícího systému. Zároveň by měl umožňoval jeho začlenění do starších systémů, které společnost nabízela dříve. A tedy umožnit komunikaci s portálem Aukro prostřednictvím nabízeného API rozhraní.

2. Cíle bakalářské práce

Cílem práce je vytvořit univerzální modul, který lze nasadit na libovolný systém provozující internetový obchod. Tento modul bude umožňovat napojení vybraných služeb aukčního serveru Aukro do administrace obchodu. Ke komunikaci mezi aukčním serverem a systémem provozujícím obchod bude použito rozhraní API, které aukční server nabízí. Prostřednictvím modulu bude možno přímo z administrace obchodu vytvářet nové aukce na tomto aukčním portále, jejich následnou správu a zpětné zakomponování úspěšných dražeb do běžných objednávek.

Součástí práce by měl být návod jak vytvořený modul zakomponovat do libovolného systému. Uvést jeho požadavky na tvůrce systému a následně na uživatele.

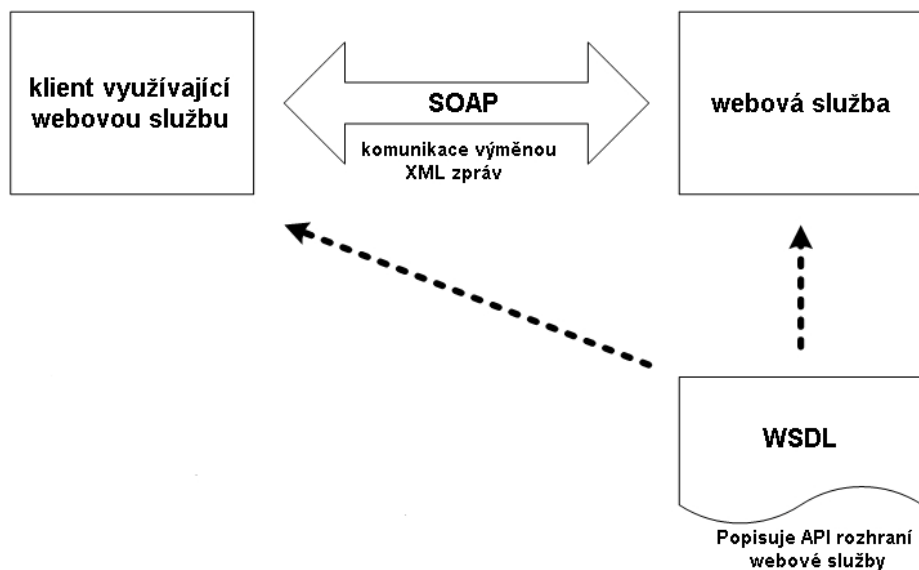
Pro prezentaci a otestování modulu vytvořit jednoduchý systém. Tento systém by měl umožňovat plné využití modulu. Tento systém následně vystavit na internet a umožnit tak testování modulu.

3. Obecné principy

3.1. API rozhraní

Jedná se o rozhraní sloužící k programování aplikací, využívající metody popřípadě třídy nějaké knihovny. Určuje, jak se jednotlivé metody volají. Mezi nejdůležitější API patří grafická (OpenGL a DirectX) a API operačních systémů (POSIX, Windows API). Pro účely této práce se však budeme zabývat tzv. webAPI.

WebAPI je rozhraní, které slouží ke komunikaci dvou systémů prostřednictvím protokolu HTTP. Je to soubor služeb, které umožňují přístup k datům jiných systému a dle možností tato data využívat popř. měnit. Protokol HTTP slouží jako aplikační vrstva pro přenos zpráv založených na jazyce XML. Samotný přenos probíhá prostřednictvím protokolu SOAP. Tento protokol lze využít v různých schématech komunikace, ale nejrozšířenější je schéma klient-server. Způsob komunikace s API rozhraním popisuje soubor jazyka WSDL. Zapisuje se pomocí XML a popisuje rozhraní webové služby, jaké nabízí funkce a jakým způsobem je lze zavolat.



Obrázek 1: schéma komunikace s rozhraním API

Zpráva v jazyce XML. Jedná se o jednoduchý XML dokument s kořenovým elementem *Envelope*. V něm jsou uzavřeny další dva prvky *Header* (hlavička) a *Body* (tělo). Hlavička není povinná, jejíž pomocí se přenáší pomocné informace pro zpracování zprávy. V těle zprávy se identifikuje volaná služba a předávají se parametry popř. návratové hodnoty.

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePriceResponse
      xmlns:m="urn:x-example:services:StockQuote">
      <Price>14.5</Price>
    </m:GetLastTradePriceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Obrázek 2: ukázka zprávy v XML jazyce

Jazyk WSDL slouží pro popis webových služeb jako množiny koncových bodů zpracovávající zprávy. Samotné operace a zprávy jsou nejprve popisovány abstraktně a poté jsou svázány s konkrétním protokolem a datovým formátem. Díky tomu lze snadno vytvořit popis rozhraní. Nejčastěji se používá pro popis služby, které si posílají zprávy prostřednictvím SOAP a HTTP.

3.2. Aukro API rozhraní

WSDL popis API aukčního systému Aukro je velice rozsáhlý a proto ho zde nebudu uvádět je dostupný na internetové adrese <http://webapi.aukro.cz/uploader.php?wsdl>. Obsahuje popis všech metod, které toto API nabízí a způsob jakým je lze volat prostřednictvím SOAP.

3.3. Tvorba komponent

Při tvorbě univerzálních komponent je důležité klást důraz především na jejich nezávislost na systémech, do kterých jsou zasazovány. Z toho vyplývá nutnost při vytváření komponenty navrhnout rozhraní, prostřednictvím kterého bude probíhat komunikace mezi komponentou a systémem. Toto rozhraní by mělo být přehledné a snadno použitelné.

Pro popis tvorby systému, které jsou složeny z komponent slouží např. návrhové vzory Structural Patterns (strukturální). Popisují obecné řešení, které se používá při návrhu aplikací. Jejich snahou je zpřehlednit systém pomocí jeho strukturizace.

4. Analýza, návrh řešení a výběr technologií

4.1. Současný stav

V tuto chvíli neexistuje nekomerční modul, prostřednictvím kterého lze přímo vystavovat produkty z elektronického obchodu na aukčním systému Aukro. A zároveň aby se dal zakomponovat do různých existujících systémů. K dispozici jsou řešení od dvou společností, které takový modul nabízí jako placené rozšíření svých komerčních systémů. Proto tato práce jistě nalezne své uplatnění.

Nabízené komerční řešení:

- oXy Online s.r.o. – nabízí modul pro svůj systém oXyShop X5
- EUROPEMC s.r.o. – umožňuje rozšíření svého systému ShopMC

4.2. Návrh řešení

Pro vývoj a testování vytvářeného modulu bude nutné vytvořit jednoduchý systém, který bude nahrazovat administraci reálného internetového obchodu.

4.2.1. Návrh modulu

Struktura modulu by měla být jednoduchá a přehledná. Díky tomu lze v budoucnu snadněji implementovat další funkce, které toto API již nabízí, a které nebudou nyní implementovány.

Vytvořený modul by měl být použitelný pro libovolný systém, proto je nutné vytvořit jednoduché rozhraní díky kterému lze předávat informace o produktech obsažené v databázi systému. Tím pádem nebude muset uživatel tato data zadávat opakovaně. A také musí být jednoduchý přístup k datům z úspěšných aukcí pro možnost vložení objednávky do systému. Vzhledem k nutnosti opakujících se informací pro každou aukci, by bylo vhodné umožnit přednastavení některých parametrů, které by se při vkládání rovnou vyplnily ve formuláři a tím se usnadnilo samotné vytváření nových aukcí. Díky tomu dojde i k eliminaci náhodného překlepu například u čísel bankovních kont.

Pro přihlášení k systému Aukro, prostřednictvím jejich API rozhraní, jsou nutné přihlašovací údaje uživatele, proto je nutné aby byla samotnému uživateli umožněna změna těchto údajů.

Modul se bude skládat ze tří částí:

1. **Nastavení modulu** – slouží k vyplnění údajů, které jsou nutné k připojení, přednastavení zvolených doprav, adres apod.
2. **Vystavení produktu** – zde lze vystavit samotný produkt v aukci
3. **Seznam vystavených** - tato část zobrazuje již vystavené produkty v aukci (probíhající, úspěšně prodané, neprodané)

4.2.2. Návrh testovacího systému

Systém bude vytvořen na podobné logice, jakou používá společnost CLWEB pro svůj CMS systém, aby následné nasazení na jejich systému bylo naprosto bezproblémové. Jedná se o modulovou strukturu. A součástí testovacího systému budou tři moduly:

- Zboží – tento modul slouží pro správu produktů
- Objednávky – zde je přehled dokončených objednávek
- Úvod – jedná se o základní modul pro zobrazení úvodní stránky

4.3. Výběr technologií

4.3.1.XHTML

Jedná se o značkový jazyk pro tvorbu webových stránek, který je vyvinutý mezinárodním konsorciem W3C. V praxi se používají tyto tři verze:

- XHTML 1.0 transitional (přechodové)
- XHTML 1.0 strict (striktní)
- XHTML 1.1

V této práci se používá verze XHTML 1.0 strict.

4.3.2.CSS

Je soubor metod pomocí kterých se upravuje grafická úprava webových stránek, která je tímto oddělena od samotného obsahu. Oddělením vzhledu od obsahu dosáhneme vyšší přehlednosti kódu.

4.3.3.JavaScript

Jedná se o interpretovaný objektově orientovaný skriptovací jazyk, který se používá pro webové stránky.

4.3.4. PHP

Je skriptovací jazyk, který pracuje na straně serveru. Je určen především pro tvorbu dynamických webových stránek. K programování lze přistupovat jak procedurálně, tak objektově orientovaným přístupem. K vytvoření této práce byla zvolena verze 5.3.5.

4.3.5. AJAX

Jedná se o obecné označení technologií, které slouží pro vývoj interaktivních webových stránek, které mění obsah (popř. část svého obsahu) bez nutnosti jejich opětovného načítání. Díky tomu jsou aplikace uživatelsky přívětivější oproti klasickým aplikacím, které tuto technologii nevyužívají. Vyžadují však moderní internetové prohlížeče. Aplikace jsou vyvíjeny použitím těchto technologií:

- (X)HTML s CSS pro prezentaci obsahu
- DOM a JavaScript pro zobrazování a dynamické změny obsahu
- XMLHttpRequest pro asynchronní výměnu dat s webovým serverem

4.3.6. MySQL

Je databázový systém, který je k dispozici jako bezplatný pod licencí GPL, ale také jako komerční s placenou licencí. Jedná se o multiplatformní, relační databázi. Komunikace s ní probíhá prostřednictvím jazyka SQL.

4.3.7. Použití open source

V modulu se využívají tyto komponenty, které jsou k dispozici na internetu pod volně šiřitelnou licencí:

FCKeditor

Jedná se o WYSIWYG textový editor, který lze použít na webových stránkách. Je velmi nenáročný a okamžitě použitelný. Je šířen pod licencemi GPL, LGPL a MPL. Jádro tohoto editoru je napsáno v jazyce JavaScript. Stránky vývojářů www.fckeditor.net

```

<?PHP

include("fckeditor/fckeditor.php");
$oFCKeditor = new FCKeditor('text') ;
$oFCKeditor->Config['AutoDetectLanguage'] = false ;
$oFCKeditor->Config['DefaultLanguage'] = 'cs';
$oFCKeditor->Width = '400';
$oFCKeditor->Height = '250';
$oFCKeditor->Value = '';
echo $oFCKeditor->CreateHtml();

?>

```

Obrázek 3: postup implementace FCKeditoru

JAK

Je objektově orientovaná framework v prostředí jazyka JavaScript. Složen z volně provázaných knihoven, které usnadňují práci. Obsahuje spousty widgetů, které se velmi snadno a rychle implementují. Je šířen pod licencí MIT a vyvíjí ho společnost Seznam.cz, a.s. Stránky frameworku <http://jak.seznam.cz/>

```

<html>
<head>
.
.
<script type="text/javascript" src="jak/jak_compressed.js"></script>
<script type="text/javascript" src="jak/cal/calendar.js"></script>
<script type="text/javascript" src="jak/cal/calendarsetup.js"></script>
.
.
</head>
<body>
.
.
<input type="text" id="datum" name="datum" size="14" value="">
<script type="text/javascript">
    JAK.Calendar.setup("jak/img/calendar_icon.gif","vybrat datum",{ pickTime: true,
        defaultFormat: ['\ 'j.n.Y H:i\ '],["datum"]});
</script>
.
.
</body>
</html>

```

Obrázek 4: postup implementace frameworku JAK

5. Realizace

5.1. Způsob realizace

Pro vytvoření systému a modulu, byly použity výše zmíněné technologie. Problém s využíváním modulu nastane pokud uživatel vypne podporu JavaScriptu ve svém prohlížeči. V tu chvíli by byl modul nefunkční. Vzhledem k tomu, že se však jedná o modul, který se nachází v administraci systému, do které má přístup omezený počet lidí stačí je s tímto omezením seznámit.

Ukázkový systém, je určen pouze pro prezentaci modulu Aukro, a pro jeho jednoduchost je programován procedurálně, pouze s využitím objektu pro komunikaci s databází. Tento objekt je instancí třídy *mysql*. Naproti tomu modul aukro je psán plně objektově orientovaným způsobem.

5.2. Logika systému

Systém pracuje na modulové bázi, kde se každý modul stará o jeden celek operací. V tomto jednoduchém ukázkovém systému jsou tři moduly, které umožňují správu zboží, objednávek a výchozí modul, který slouží jako uvítací stránka.

Každý modul musí mít stejnojmenný adresář v adresáři moduly. Použití jednotlivých modulů se řídí pomocí proměnné *modul*, která se definuje v souboru *config.php* podle hodnoty předané metodou *get* (v odkazu), nebo *post* (prostřednictvím formuláře). Dle této proměnné se vloží hlavní skript modulu *main.php* a ten určí jaké další soubory budou vloženy na základě předávané hodnoty proměnné *what*.

5.2.1. Modul zboží

Tento modul slouží pro správu zboží. Umožňuje vkládání nových produktů do systému, jejich výpis, editaci a mazání. Při vkládání je možné nahrávat rovnou obrázky ze kterých se, pro zvýšení výkonu systému, dělají zmenšeniny pro zobrazení v katalogu. Pro vložení produktu se proměnná *what* nastaví na *new* a vloží se skript ze souboru *form.php*, který zobrazí formulář pro vyplnění jednotlivých informací o produktu. Odesláním formuláře je proměnná nastavena na *save* a zavolán skript ze souboru *save.php*, který provede samotné vložení, popřípadě editaci dat v databázi. Také volá funkci, která provede zmenšení nahraných obrázků a jejich uložení. Pro vylistování zboží ze systému se proměnná nastaví na *list*, v tu chvíli se vloží soubor *list.php* a provede samotné zobrazení seznamu produktů. Ze

seznamu je možné přejít na editaci produktu. V tom případě je proměnná nastavena na *edit* a je vložen soubor *read.php*, který načte jednotlivé informace o produktu a opět je vložen soubor *new.php* s formulářem, ve kterém jsou informace zobrazeny s možností editace. Po odeslání následuje stejný postup jako při vložení nového produktu.

5.2.2. Modul objednávky

Modul, který slouží pro výpis objednávek ze systému. Umožňuje pouze celkový výpis všech objednávek a zobrazení detailu jednotlivých objednávek. Pro výpis je proměnná *what* nastavena na *list* a je vložen skript *list.php*, který provede vypsání seznamu objednávek s možností přejít na detail vybrané objednávky. Přičemž se proměnná nastaví na *detail* a je vložen soubor *detail.php*, který obsahuje skript pro podrobný výpis zvolené objednávky.

5.2.3. Modul úvod

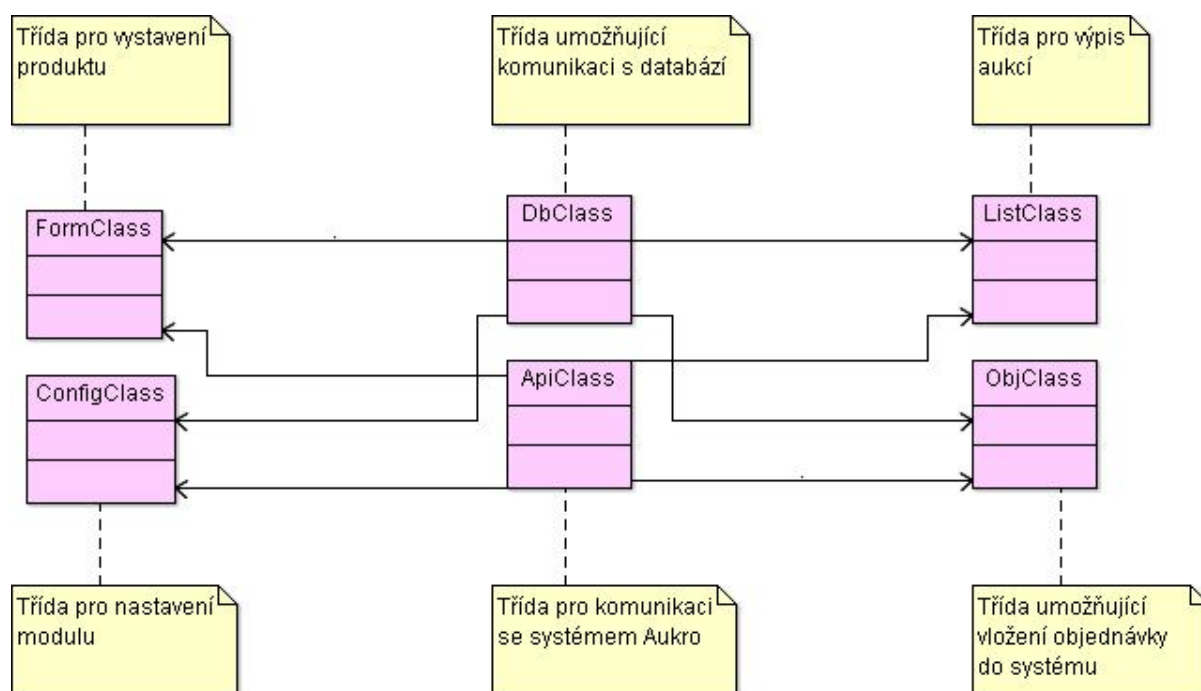
Tento modul je určen k zobrazení uvítací stránky, na které jsou uvedené informace o systému a omezení pro jeho testování. Vždy je vložen soubor *info.php*, který obsahuje zmíněné informace.

5.3. Logika modulu Aukro

Tento modul pracuje na stejné bázi jako ostatní moduly systému. To znamená, že je umístěn ve stejnojmenném adresáři v adresáři moduly. A jeho hlavním skriptem je soubor *main.php*, který rozhoduje o vkládání dalších souborů. Jedné ze tří částí modulu Tyto části jsou: nastavení modulu, vystavení produktu a zobrazení seznamu již vystavených produktů na aukčním portále Aukro.

Komunikace se systémem Aukro probíhá prostřednictvím protokolu SOAP. Výměna informací mezi modulem a aukčním systémem probíhá prostřednictvím jedné třídy, která obsahuje vybrané funkce, jenž tento systém nabízí prostřednictvím svého API.

Přihlašovací údaje nutné pro přihlášení a vystavování produktů jsou uloženy v databázi. Ke komunikaci s databází slouží jedna třída, která zajišťuje všechny dotazy potřebné pro správnou funkci modulu.



Obrázek 5: schéma propojení tříd modulu

5.3.1. Část 1 – nastavení modulu

Pro nastavení modulu je řídící proměnné modulu *what* přiřazena hodnota *nastaveni*, tím je vložen skript ze souboru *nastaveni.php*. V něm se po vložení třídy *Config* ze souboru *ConfigClass.php*, vytvoří její instance. Voláním metody *getMenu* je vráceno menu, které umožňuje přechod mezi jednotlivými skupinami nastavení. Následně se volá metoda *getFormByChance*, a ta dle výběru části nastavení vrací formulář. Při vytváření instance této třídy je třeba konstruktoru předat platné spojení k MySQL databázi, protože se zároveň vytvoří vnitřní instance třídy *Db*, která pracuje s databází. Prostřednictvím tohoto objektu se vložená data ukládají do tabulky, kterou si v případě, že se ještě v databázi nenachází, vytvoří.

Sloupec	Typ	Porovnávání	Vlastnosti	Nulový	Výchozí	Extra
co	varchar(50)	utf8_czech_ci		Ne	Žádná	
hodnota	text	utf8_czech_ci		Ne	Žádná	

Obrázek 6: struktura databázové tabulky

Nastavení je rozděleno do tří celků. První slouží pro nastavení přihlašovacích údajů, která jsou nutná pro přihlášení k serveru. Druhý celek nastavení je určen pro předdefinování doprav, které obchod používá pro doručování zboží k zákazníkům a předvyplnění bankovních kont. Třetí část nastavení je určena pro předdefinování místa, kde se daný obchod nachází. Tyto údaje jsou vyžadovány při vystavování produktů. Druhou a třetí část nastavení lze

vyplňovat a upravovat až po vyplnění platných údajů v první části, protože tyto části vyžadují již komunikaci se serverem Aukro. Ta je zajištěna prostřednictvím třídy *Api* ze souboru *ApiClass.php*. Tato třída zajišťuje komunikaci mezi modulem a aukčním serverem Aukro.

5.3.2. Část 2 – vystavení produktu

Pro možnost vystavení produktu se proměnná *what* nastaví na *new* a je vložen soubor *new.php*. Tento soubor zajišťuje vytvoření instance třídy *Aukro_form* ze souboru *FormClass.php*. Jsou dvě možnosti jak může být tato instance vytvořena, záleží zda je předáno id produktu, který se nachází v obchodě, nebo jestli se vkládá produkt, který v obchodě není. Předané id je nutné pro zpětné spárování produktů mezi objednávkou ze systému Aukro a zbožím v obchodě. Tato třída využívá instance dvou jiných tříd. A sice třídu, která slouží pro komunikaci mezi modulem a systémem Aukro (třída *Api*) a další třídu, která slouží pro komunikaci s databází (třída *Db*). Objekty těchto tříd se vytvářejí přímo v konstruktoru, kterému je nutné předat platné spojení k MySQL databázi.

Pokud je id produktu předáno, vytvoří se instance s tímto lokálním identifikátorem. A je možné tomuto objektu předat data, která jsou o produktu vyplněna v obchodě a lze je použít. K tomu slouží nastavovací metody:

- *setName* – slouží k nastavení názvu produktu
- *setPopis* – nastavuje popis produktu
- *setPoc_cena* – nastaví hodnotu počáteční ceny dražby
- *setVyh_cena* – definuje vyhrazenou cenu aukce
- *setKup_cena* – nastaví cenu za kterou je možno okamžitě zakoupit produkt
- *setDirectory* – nastaví adresář kde se nachází obrázky k produktu
- *setObrázky* – nastaví název souboru s obrázkem daného produktu

Není nutné využít všechny metody. Předaná data jsou následně předvyplněna do formuláře, který je vygenerován pomocí metody *createForm*. Tato metoda se volá i v případě, že není předáno id a je vytvořena prázdná instance. Formulář obsahuje také předvyplněné informace z nastavení modulu. Jedná se o předdefinované dopravní možnosti, čísla bankovních kont a místo kde se obchod nalézá.

Po vyvolání metody *createForm* je zobrazena první část formuláře, kde se volí kategorie daného produktu. Výběr probíhá ze seznamu, který vrátil systém Aukro, pomocí technologie AJAX pro rychlejší zvolení správné kategorie do které produkt patří. Po vybrání

některé z koncových kategorií se zobrazí její id v textovém poli a umožní se přejít na druhou část formuláře. V té se kromě povinných polí, které musí uživatel vyplnit při každé aukci, načtou i podrobné informace o produktu, které se liší na základě zvolené kategorie. Formulář je rozdělen do čtyř záložek, mezi kterými se přepíná pomocí funkce v jazyce JavaScript. Ta vždy zobrazí pouze aktuální záložku a ostatní skryje. Záložky, do kterých je formulář rozdělen s krátkým popisem jejich obsahu:

- Popis a fotografie – v této záložce jsou pole pro vyplnění názvu produktu, jeho popis a obrázek, uvádí se počet kusů a především záložka obsahuje parametry na základě zvolené kategorie
- Cena a čas – tato záložka je určena pro vyplnění cen produktu, čas kdy má aukce započít a délka trvání aukce
- Doprava a platba – záložka je určena pro vyplnění informací o dopravě produktu, tyto informace lze předvyplnit v nastavení a není nutno je upravovat, součástí záložky jsou i čísla bankovních kont a lze připsat poznámku k dopravě
- Ostatní – záložka obsahující umístění obchodu a nadstandardní volby, které umožňují zviditelnit produkt, ale ty jsou zvlášť zpoplatněny

Po vyplnění všech potřebných údajů se formulář odešle a dle jednotlivých typů parametrů se vygeneruje pole obsahující všechny volby. Toto pole se odešle prostřednictvím instance třídy *Api* na aukční server, kde dojde k jeho zpracování a vrátí pole, které obsahuje přidělené id v rámci systému Aukro a cenu poplatku za vystavení. Vracené id se spolu s id, pod kterým je produkt v systému, uloží prostřednictvím objektu třídy *Db* do tabulky, jejíž existenci při tom ověří a popřípadě ji vytvoří. Tato tabulka slouží následně pro zpětné spárování produktů.

5.3.3. Část 3 – seznam vystavených

Pro zobrazení seznamu vystavených aukcí je proměnná *what*, která řídí vkládání jednotlivých souborů, nastavena na hodnotu *list* a je vložen soubor *list.php*, který vytvoří instanci třídy *Lists* a zavolá její metodu *getList*. Při vytváření objektu je nutno předat konstruktoru platné spojení k MySQL databázi, protože se zároveň vytváří vnitřní instance třídy *Db*. Současně se v konstruktoru vytvoří instance třídy *API*, která slouží pro komunikaci mezi modulem a aukčním systémem.

Jsou možné tři druhy výpisu podle aktuálního stavu aukce produktů. A dle toho jsou dispozici i různé akce, které může uživatel provést. Druhy výpisů:

- Vystavené – jedná se o probíhající aukce, ve kterých mohou zákazníci přihazovat
- Prodané – v tomto výpisu jsou aukce, které skončily a byl o ni zájem
- Neprodané – tento výpis zobrazuje aukce, které skončily a nikdo o ně neprojevil zájem

Vystavené

V tomto výpisu jsou aukce, které právě probíhají. Uživatel může vybranou aukci ukončit. Pokud již v aukci bylo přihazováno, dokončením se aukce uzavře a výhercem je poslední přihazující a aukce se přesune do prodaných. Pokud v aukci nikdo neprojevil o produkt zájem je aukce skončena a přesunuta do neprodaných.

Prodané

Zde jsou úspěšné aukce, které skončily v řádném termínu, byli předčasně ukončeny s platným přihozením, nebo skočili zakoupením za cenu "kup teď". Je zde možnost podívat se na detail kupujícího prostřednictvím vyskakujícího okna, které je vyvoláno pomocí JavaScriptu. Dále je možnost přejít na souhrnná data (detail) dražby, kde se zobrazí, kupujícím zvolený způsob dopravy, jeho kontaktní údaje a souhrnná cena za zboží včetně ceny dopravy. Zde je možnost vytisknout si tento detail, který se otevře v novém okně a po načtení je rovnou zobrazeno dialogové okno pro tisk. Také je zde možnost vložit si tato data do objednávek v systému. K tomu složí soubor *insert_obj.php*, ve kterém se vytvoří instance třídy *Obj*. Tato třída slouží pouze pro načtení dat dokončené aukce a prostřednictvím návratových metod vrací potřebné hodnoty, využívá instance tříd *Api* a *Db*. Po vytvoření objektu této třídy se zavolají dvě metody. První *readObj*, které je nutno předat id aukce, slouží k načtení dat vybrané aukce a při úspěchu vrací *true*. Druhá *getLocalId* se pokusí najít id, pod kterým je daný produkt v systému podle id aukce, pokud se podaří najít vrátí funkce *true*. Po úspěšném vykonání obou funkcí je již možnost volat návratové funkce.

Návratové funkce jsou tyto:

- | | |
|---------------------|-------------------------|
| ▪ <i>getJmeno</i> | ▪ <i>getCenaDopravy</i> |
| ▪ <i>getUlice</i> | ▪ <i>getPoznamka</i> |
| ▪ <i>getMesto</i> | ▪ <i>getDatum</i> |
| ▪ <i>getPsc</i> | ▪ <i>getLocalId</i> |
| ▪ <i>getTel</i> | ▪ <i>getCenaPolozky</i> |
| ▪ <i>getEmail</i> | ▪ <i>getPocet</i> |
| ▪ <i>getDoprava</i> | |

Neprodané

V tomto přehledu jsou k dispozici aukce, které skončily v řádném termínu, nebo které ukončil uživatel a nebyl o daný produkt projevem zájem. Je zde umožněno vybraný produkt vystavit znovu, pouze po vyplnění údajů o délce aukce a jejím začátku. Při znovuvystavení produktu lze tuto neúspěšnou aukci rovnou smazat. K dispozici je možnost nechat si zaslat potvrzení o vystavení na email.

5.4. Adresářová struktura systému

Adresářová struktura je následující:

- **config/** - adresář konfiguračních souborů
- **css/** - adresář s css stylem a souborem funkcí v jazyce JavaScript
- **css/image/** – adresář obsahující obrázky vzhledu
- **fckeditor/** - adresář obsahující wisiwig editor
- **jak/** - adresář obsahující framework v jazyce JavaScript
- **moduly/** - adresář obsahující jednotlivé moduly
- **moduly/aukro** – adresář modulu umožňující práci s aukčním systémem Aukro
- **moduly/uvod** – adresář se základním modulem
- **moduly/objednavky** – adresář modulu obsluhujícího objednávky
- **moduly/zbozi** – adresář modulu pro správu zboží
- **picture/** - adresář s obrázky pro zboží

5.5. Adresářová struktura modulu aukro

Modul se nachází v adresáři *moduly/aukro/*. A jeho struktura je následující:

- **classes** – adresář obsahující jednotlivé třídy modulu
- **css** – adresář s css stylem pro zobrazení výstupů z modulu
- **js** – adresář se skripty v jazyce JavaScript

5.6. Souborová struktura systému

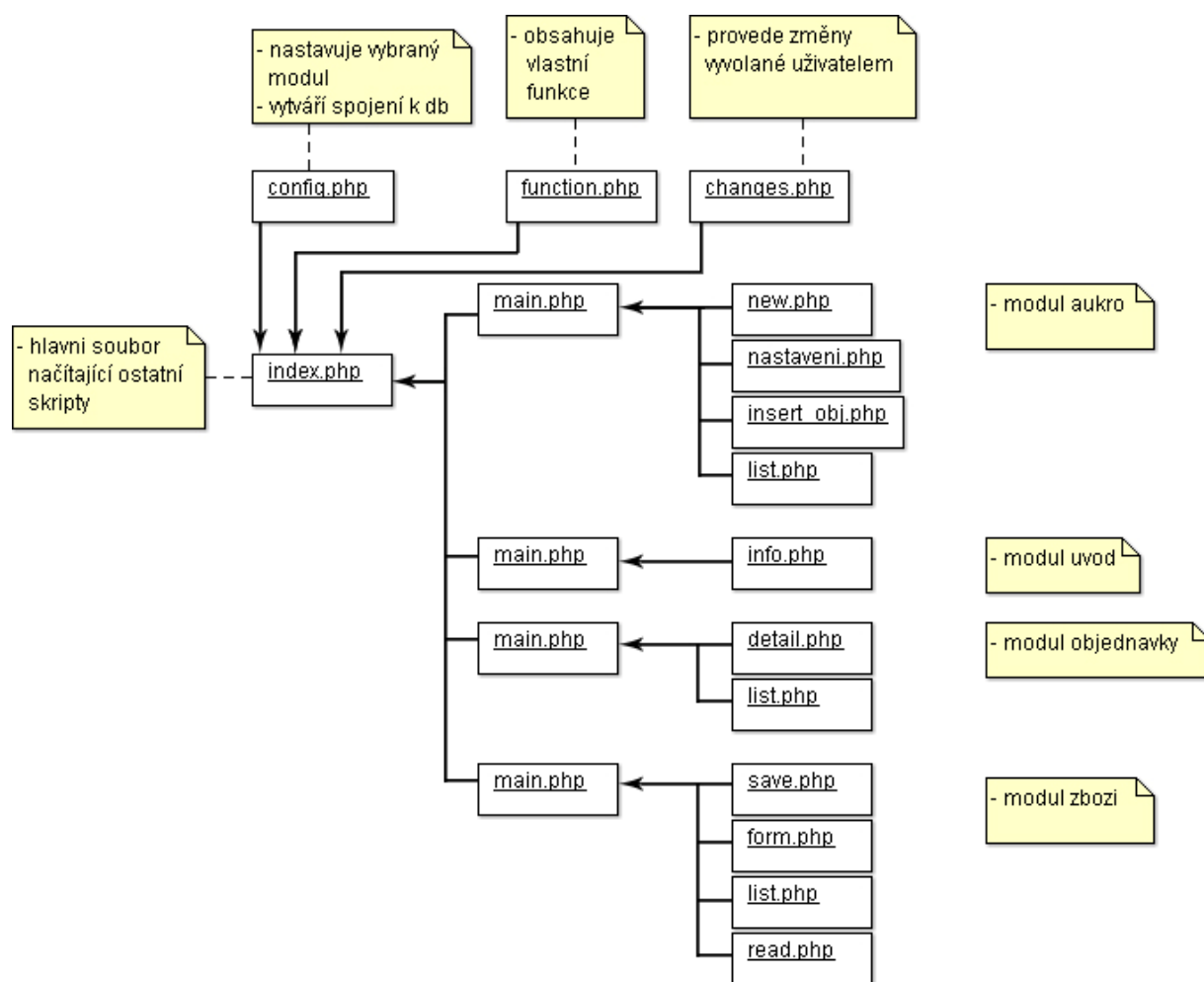
Hlavním souborem, který se vždy vykoná je `index.php`, který se nachází v kořenovém adresáři. Stará se o načítání všech souborů, které jsou aktuálně potřebné. Obsahuje jako jediný html hlavičku. Další důležité soubory s umístěním a popisem:

- **index.php** – hlavní soubor načítající potřebné skripty
- **config/config.php** – konfigurační skript, nastavuje aktuální modul, vytváří spojení s databází
- **config/function.php** – soubor obsahující vlastní funkce
- **config/changes.php** – skript obsluhující změny vyvolané uživatelem
- **css/function.js** – soubor obsahující JavaScriptové funkce
- **css/style.css** – soubor s css stylem pro vzhled systému
- **fckeditor/fckeditor.php** – hlavní soubor fckeditoru
- **jak/jak_compressed.js** – soubor s jádrem frameworku JAK
- **moduly/objednavky(uvod, zbozi)/main.php** – stará se o vložení skriptů modulu
- **moduly/objednavky(zbozi)/list.php** – skript sloužící k vylistování objednávek (zboží)
- **moduly/objednavky/detail.php** – skript pro zobrazení detailu objednávky
- **moduly/zbozi/form.php** – skript sloužící k zobrazení formuláře pro vložení a editaci produktu
- **moduly/zbozi/read.php** – skript, který složí pro načtení informací o produktu z databáze
- **moduly/zbozi/save.php** – skript pro uložení produktu do databáze

5.7. Souborová struktura modulu

Modul aukro je koncipován stejně jako ostatní moduly systému. Hlavní skript *main.php* se stará o vložení správného skriptu dle aktuální potřeby. Modul se skládá z těchto souborů:

- **classes/ApiClass.php** – třída sloužící pro komunikaci s API systému Aukro
- **classes/ConfigClass.php** – třída, která umožňuje nastavení modulu
- **classes/DbClass.php** – třída obsluhující komunikaci s databází v rámci modulu
- **classes/FormClass.php** – třída pro zobrazení formuláře a vystavení produktu na portále Aukro
- **classes/ListClass.php** – třída pro práci s již vystavenými produkty
- **classes/ObjClass.php** – třída která vrací informace o dokončených aukcích, pro možnost vytvoření objednávky v systému
- **css/style.css** – soubor obsahující css styl pro vzhled výstupu z modulu
- **js/function.js** – soubor obsahující funkce v jazyce JavaScript pro potřeby modulu
- **insert_obj.php** – skript vkládající objednávku do systému
- **list.php** – skript, který zobrazuje již vystavené produkty
- **main.php** – hlavní skript modulu, který rozhoduje o vložení dalších skriptů modulu dle potřeby
- **nastaveni.php** – skript zobrazující možnosti nastavení
- **new.php** – skript obsluhující vystavení nového produktu na systému Aukro
- **podCat.php** – skript volaný prostřednictvím technologie AJAX, pro výběr kategorie systému Aukro
- **tisk.php** – slouží pro vytvoření tisknutelného detailu dokončené dražby
- **user_detail.php** – slouží k zobrazení detailu uživatele ve vyskakujícím okně



Obrázek 7: diagram vkládání skriptů v systému

5.8. Popis php skriptů

Každý soubor obsahující php skripty má na svém začátku hlavičku, která obsahuje jméno autora a je zde popsána funkcionality daného souboru. Samozřejmě nechybí ani komentáře u jednotlivých metod, u kterých jsou popsány vstupní parametry, výstup dané metody a jednotlivé operace, které nejsou na první pohled zřejmé.

```

/*****
 *                               Modul Aukro                               *
 *****/
 *
 * Trida slouzici pro zobrazeni vystavenych, prodanych a neprodanych produktu.
 * Vyuziva instance tridy ApiClass a DbClass.
 * Pro vytvoreni instance teto tridy je nutne predat konstruktoru platne spojeni
 * k MySQL databazi.
 *
 * @copyright Copyright (c) 2011 Martin Hozak
 *
 * @version 1.0
 */

```

Obrázek 8: ukázka komentáře

6. Postup nasazení modulu

Tento modul je od začátku tvořen tak, aby bylo umožněno jeho snadné zasazení do libovolného systému, proto se tato kapitola bude věnovat nasazení modulu na již existující systém. Jsou dvě možnosti jak toho dosáhnout. První je, že se do systému převezme celá struktura modulu použitím souboru *main.php*. Druhou možností je použití jednotlivých částí dle potřeb systému.

Při zvolení jakékoliv ze dvou možností je nutné splnit ještě následující požadavky. Složka FCKeditoru musí být umístěna v kořenovém adresáři stránek. Pro snadný výběr času zveřejnění aukce je použit widget z JavaScriptového frameworku JAK. Složka s tímto frameworkem může být umístěna kdekoli, ale soubory frameworku musí být správně vloženy do hlavičky stránek. Do hlavičky také musí být vložen kaskádový styl samotného modulu, který se nalézá v adresáři *css*. V adresáři *js* je soubor *function.js*, který obsahuje funkce v jazyce JavaScript.

```
<script src="moduly/aukro/js/function.js" type="text/javascript"></script>
<link rel="stylesheet" type="text/css" href="moduly/aukro/css/style.css">

<link rel="stylesheet" href="jak/css/calendar.css" media="all" type="text/css">
<script type="text/javascript" src="jak/jak_compressed.js"></script>
<script type="text/javascript" src="jak/cal/calendar.js"></script>
<script type="text/javascript" src="jak/cal/calendarsetup.js"></script>
```

Obrázek 9: ukázka vložení nutných souborů do hlavičky

6.1. Převzetí celé struktury

Pro ovládání modulu se bude využívat soubor *main.php* se skriptem pro vkládání jednotlivých částí. Budou se využívat dvě řídicí proměnné a sice proměnná *modul* a proměnná *what*. V tomto případě musí být v systému zajištěno, aby vždy když je proměnná *modul* nastavena na aukro, byl vložen soubor *main.php*, který se postará o další vkládání potřebných skriptů. Pro správnou funkčnost modulu bude nutné upravit předávání dat o produktu a vkládání objednávek. Tomu se bude věnovat jedna z následujících kapitol.

6.2. Vkládání jednotlivých částí modulu

Při zvolení tohoto způsobu nastavení nejsou potřeba soubory, které umožňují nasazení na testovacím systému, lze z nich však zkopírovat kód pro spuštění modulu na vlastních stránkách. Jedná se o tyto soubory:

- list.php
- main.php
- nastaveni.php
- new.php

6.2.1. Zprovoznění nastavení

Nastavení modulu je instance třídy *Config*, proto je nutné do stránky, kde chceme mít nastavení tohoto modulu vložit tuto třídu a následně volat její dvě metody, které nám zobrazí menu a formuláře pro vyplnění. Jedinou nutnou úpravou kódu je změna odkazů v menu, které ukazují na jednotlivé části nastavení. Jedná se o metodu *getMenu* kde je nutno tyto odkazy upravit, aby odkazovaly na stránku, kam vkládáme nastavení. Předávané proměnné *modul* a *what* nadále nebudou potřeba, ale proměnná *subwhat* se musí stále pomocí odkazu předávat.

Na obrázku 8 je vidět vložení nastavení do libovolné stránky. Důležité je předání platného spojení k MySQL databázi při vytváření instance třídy. Dále je volána metoda *getMenu* pro vytvoření menu, sloužící pro přechod mezi jednotlivými skupinami nastavení. Metoda *getFormByChance* vrátí, podle hodnoty proměnné *subwhat*, požadovaný formulář.

```
include ("./moduly/aukro/classes/ConfigClass.php");
$nastaveni=new Config($mysqli);
echo $nastaveni->getMenu();
echo $nastaveni->getFormByChance();
```

Obrázek 10: ukázka vytvoření stránky pro nastavení

Obrázek 9 ukazuje změnu odkazu v metodě *getMenu*, v případě, že je nastavení vloženo do skriptu v souboru *nasazeni.php*.

```
<a href="?modul=aukro&what=nastaveni&subwhat=basic"> Nastavení přístupu</a>
<a href="nasazeni.php?subwhat=basic"> Nastavení přístupu</a>
```

Obrázek 11: ukázka změny adresy odkazu

6.2.2. Zprovoznění vystavování aukcí

K vystavení produktu slouží třída *Aukro_form*. Po jejím vložení do skriptu se vytvoří její instance s předaným platným spojením k databázi a pokud se budou vkládat do objektu informace o produktu tak také id, pod kterým je produkt v systému, pro možnost zpětného spárování. Pokud se jedná o produkt, který není v systému, místo id se použije libovolné číslo, které nebude kolidovat s možným identifikátorem produktu. Po vytvoření objektu lze předávat pomocí nastavujících metod informace o produktu. Na závěr se zavolá metoda *creatForm*, která vrátí již formulář pro vystavení aukce.

```
include ("./moduly/aukro/classes/FormClass.php");

if (@$_GET['id']!=''){
    $form = new Aukro_form($_GET['id'],$mysqli);
    $form -> setName($nazev['hodnota']);
    $form -> setPopis($popis['hodnota']);
    $form -> setPoc_cena($cena['hodnota']);
    $form -> setVyh_cena(0.00);
    $form -> setKup_cena($cena['hodnota']);
    $form -> setDirectory('./picture/ico');
    $form -> setObrazky($obrazky['top_obrazek']);
}else{
    $form = new Aukro_form(1,$mysqli);
}
echo $form -> creatForm();
```

Obrázek 12: ukázka implementace vystavování produktů

6.2.3. Zprovoznění seznamu aukcí

K výpisu aukcí slouží třída *Lists*, proto je nutné nejdříve vložit tuto třídu do stánek a následně vytvořit její instanci předáním platného spojení k databázi. Po vytvoření objektu se zavolá metoda *getList*, která vrátí již samotný výpis.

```
include ("classes/ListClass.php");
$list= new Lists($mysqli);

echo $list->getList();
```

Obrázek 13: ukázka implementace seznamu vystavených produktů

Pro funkčnost jednotlivých akcí, které může uživatel s aukcemi vykonávat je nutné změnit cesty v odkazech. První metoda, kterou je nutno upravit je *getMenu*, která umožňuje přechod mezi jednotlivými skupinami aukcí. V těchto odkazech je nutno zanechat a nadále předávat proměnnou *subwhat*. Další odkaz, který je nutno upravit, aby směřoval na platné

umístění seznamu aukcí, se nachází v metodě *getVystavene*, pomocí tohoto odkazu se předává id aukce, která má být dokončena. V odkazu musí zůstat proměnné *subwhat* a *zrusit*. Metoda v které se také musí změnit odkazy se jmenuje *getProdane*. Zde je nutné změnit odkazy na detail dokončené aukce. Proměnné *modul* s *what* se mohou odstranit, musí však zůstat proměnné *subwhat* a *id*. V této metodě se musí ještě změnit adresa odkazující na umístění souboru *user_detail.php*. V metodě pro zobrazení detailu aukce s názvem *getProdaneDetail* je nutno změnit umístění souboru *tisk.php*, který slouží pro tisk detailu aukce. Další odkaz, který se nachází v této třídě a je nutné ho změnit, odkazuje na soubor, který umožňuje vkládání do objednávek a součástí odkazu musí zůstat proměnná *id*, která předává id vkládané aukce. V souboru pro vkládání objednávek do systému musí být vložena třída *Obj* a vytvořena její instance předáním platného spojení k databázi. Následně je nutno zavolat metodu *readObj*, jejímž parametrem je id aukce a metodu *getLocalId*, která spáruje vydražený produkt s produktem v systému. Voláním návratových metod jsou vrácena data z aukce a lze je rovnou vložit do databáze.

```
include ("./moduly/aukro/classes/ObjClass.php");
$obj = new Obj($mysqli);

if (($obj->readObj($_GET['id'])) and ($obj->getLocalId())) {
    $dotaz="INSERT INTO objednavky (jmeno,ulice,mesto,psc,tel,email,doprava,cena_dopravy,
                                poznamka,datum)
                                VALUES ('".$obj->getJmeno()."', '".$obj->getUlice()."',
                                '".$obj->getMesto()."', '".$obj->getPsc()."',
                                '".$obj->getTel()."', '".$obj->getEmail()."',
                                '".$obj->getDoprava()."', '".$obj->getCenaDopravy()."',
                                '".$obj->getPoznamka()."', '".$obj->getDatum()."'");
    $mysqli->query($dotaz);
    $obj_id=$mysqli->insert_id;
    if($obj_id>0){
        $dotaz="INSERT INTO polozky (objednavka,zbozi,cena,pocet)
        VALUES ('".$obj_id."', '".$obj->getLocalId()."',
        '".$obj->getCenaPolozky()."', '".$obj->getPocet()."'");
        $mysqli->query($dotaz);
    }
}
```

Obrázek 14: ukázka skriptu vytvářející objednávky

Poslední metodou, ve které je potřeba něco upravit je *getNeprodane*, zde je nutné změnit odkaz, prostřednictvím kterého se přechází na možnost znovu vystavení aukce. Odkaz by měl směřovat na list aukcí a musí zůstat zachovány proměnné *subwhat* a *id*.

6.3. Oživení modulu

6.3.1. Požadavky k systému Aukro

Pro funkci modulu je nutné mít vytvořený účet v systému Aukro, který po přijetí potvrzovacího dopisu je nutné aktivovat. Následně je nutno požádat o vygenerování tzv. API klíče, který je pro každého uživatele unikátní. Jsou dva druhy klíče, jeden je zdarma a druhý je zpoplatněn a po zaslání platí půl roku, poté je nutno jeho platnost prodloužit. Současná verze modulu vyžaduje pouze základní klíč, který je zdarma.

6.3.2. Nastavení modulu

Aby modul fungoval musí se nastavit přihlašovací informace uživatele. K tomu slouží v nastavení záložka *Nastavení přístupu*. Zde se nastaví uživatelské jméno a heslo do systému Aukro. Dále je třeba uvést email, pod kterým je uživatel na Aukru registrován, zasláný API klíč. Země a URL API jsou vzájemně provázány, protože společnost provozující aukční server Aukro provozuje podobné systémy i v dalších zemích a proto je nutné zadat vždy správnou adresu serveru a k tomu správný kód, pod kterým je daná země zavedena u provozovatele systému Aukro.

Uživatel:	<input type="text" value="jmeno"/>
Heslo:	<input type="password" value="hesloheslo"/>
Email:	<input type="text" value="jmeno@domena.cz"/>
API key:	<input type="text" value="1c7ac0a226"/>
Země:	<input type="text" value="56"/>
URL API:	<input type="text" value="http://webapi.aukro.cz/uploader.php?wsdl"/>

Obrázek 15: ukázka nastavení přihlašovacích údajů

Další nutné nastavení, které lze provést až po vyplnění formuláře v záložce *Nastavení přístupu*, je v *Nastavení adresy*. Zde se vyplní město, poštovní směrovací číslo a region. Vše se opět musí shodovat s nastavením v systému Aukro. V tuto chvíli lze již plnohodnotně využívat modul k práci s aukčním systémem.

Město	<input type="text" value="Liberec"/>
PSČ	<input type="text" value="460 15"/>
Region	<input type="text" value="Liberecký"/>

Obrázek 16: ukázka nastavení umístění obchodu

Poslední část nastavení slouží pro usnadnění práce při vystavování produktů. Lze ji vyplnit až po vyplnění záložky *Nastavení přístupu*. Definují se zde čísla bankovních kont a způsoby doprav, které se nabízejí pro dopravu zboží k zákazníkovi.

Doprava	<input type="radio"/> Prodejce platí za dopravu		
	<input checked="" type="radio"/> Kupující platí za dopravu		
Bankovní účet 1	<input type="text" value="123456789/1234"/>		
Bankovní účet 2	<input type="text" value="987654321/4321"/>		
<input checked="" type="checkbox"/>	Obyčejný balík (první položka)	<input type="text" value="80"/>	Kč
<input type="checkbox"/>	Obchodní balík (první položka)	<input type="text"/>	Kč
<input type="checkbox"/>	Cenný balík (první položka)	<input type="text"/>	Kč
<input type="checkbox"/>	EMS (první položka)	<input type="text"/>	Kč
<input type="checkbox"/>	Obyčejné psaní (první položka)	<input type="text"/>	Kč
<input type="checkbox"/>	Cenné psaní (první položka)	<input type="text"/>	Kč
<input type="checkbox"/>	Doporučená zásilka (první položka)	<input type="text"/>	Kč
<input type="checkbox"/>	Dobírka (první položka)	<input type="text"/>	Kč
<input checked="" type="checkbox"/>	Kurýrní služba (první položka)	<input type="text" value="100"/>	Kč
<input checked="" type="checkbox"/>	Kurýrní služba - dobírka (první položka)	<input type="text" value="150"/>	Kč
<input type="checkbox"/>	Zásilka do zahraničí (první položka)	<input type="text"/>	Kč

Obrázek 17: ukázka nastavení doprav

6.4. Prezentace modulu

Modul je zprovozněn na testovacím systému, které byl speciálně pro tento účel vytvořen. Jeho pomocí si lze vyzkoušet funkce modulu. Celý systém i s modulem je vystaven na internetové adrese <http://bp.leela.clweb.cz>

7. Závěr

Na základě požadavků společnosti CLWEB jsem vytvořil modul, který lze začlenit do systému a tím využívat vybrané funkce aukčního portálu Aukro.cz. Je obsažena kompletní funkcionalita, kterou společnost požadovala. Základní koncepce modulu byla tvořena s ohledem na modulovou strukturu, kterou tato společnost používá. A lze ho tak bez problémů zasadit do jejich nabízeného CMS systému. Pomocí drobných úprav ho však lze nasadit na jakémkoliv systému.

Zasazení modulu do systému bylo popsáno v šesté kapitole, této práce, která se věnuje tomuto problému. Její součástí je i postup nastavení modulu, aby byl plně funkční a umožnit využívání služeb aukčního systému Aukro.

Vývoj tohoto modulu bude záviset především na požadavcích společnosti CLWEB a jejich zákazníků. Jako další možnosti rozvoje modulu, bych viděl rozšíření vybrané sady funkcí, nabízených prostřednictvím API rozhraní. Implementovat možnost využití jiných databázových systémů. Možnost uživatelem definovat šablony pro vzhled různých výrobků dle vybrané kategorie.

8. Zdroje informací

- [1] ŠKULTÉTY Rastislav. JavaScript Programujeme internetové aplikace.
Praha: Computer Press a.s., 2004. ISBN 80-251-0144-4
- [2] GUTMANS, A - BAKKEN, S - RETHANS, D. Mistrovství v PHP 5. Praha: Computer Press a.s., 2005. ISBN 978-80-251-1519-0
- [3] Aukro API documentation [online].
URL: < <http://webapi.aukro.cz> >.
- [4] Oficiální stránky MySQL databáze [online].
URL: < <http://www.mysql.org> >.
- [5] Dokumentace k PHP [online].
URL: < <http://www.php.net/docs.php> >.
- [6] Stránky o PHP programování [online].
URL: < <http://php.vrana.cz> >.
- [7] Stránky o PHP a XML [online].
URL: < <http://www.kosek.cz> >.

Příloha – A obsah přiloženého CD

Přiložené CD obsahuje adresáře s daty:

- Systém – kompletní zdrojové kódy ukázkového systému se začleněným modulem
- Fckeditor – wisiwig textový editor
- Jak – JavaScript framework
- AukroModul – zdrojové kódy samostatného modulu
- BakalářskáPráce – dokument bakalářské práce ve formátu PDF